NRL Report 7348

UNCLASSIFIED

Another Associative Processor

JOHN E. SHORE

Information Systems Group
Office of the Associate Director of Research
for Electronics

and

TERRY L. COLLINS

Intercept and Signal Processing Branch
Communications Sciences Division

December 30, 1971

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

UNCLASSIFIED

CONTENTS

	Page
Abstract	iii
Problem Status	iii
Authorization	iii
Acronyms	iv
1. Introduction and Summary	1
2. Word Description	2
2.1 Multifield Search	2
2.2 Control of Word Contents	3
2.3 Control of Word Output	4
3. Hardware Implementation	4
3.1 Notation and Definitions	5
3.2 Equations	6
3.3 Chip Description	7
3.4 Word Construction	7
3.5 Multiple Use of Chips	7
4. System Description	8
4.1 Vertical Control	8
4.2 Horizontal Control	9
4.3 Response Store Structure	9
5. Examples of AP Operations	12
5.1 Search for the Word With the Greatest Contents	12
5.2 Multiplication of Fields in Adjacent Words	13
5.3 Column Addition	14
6. The Associative Processor Controller	14
7. Future Work	16
8. Acknowledgment	16
REFERENCES	16

UNCLASSIFIED

ABSTRACT

The logical design of an associative processor (AP), which has evolved from the design previously reported in NRL Report 6961, has now been oriented toward the requirements of the Advanced Avionic Digital Computer (AADC) under development by the Naval Air Systems Command. The AP is a byte-variable processor with multifield search, arithmetic, and logical capability. Data flow within the AP is a vertical trickle. Data for each word is fed in from the word above and fed out to the word below. The output may be the same as the input (functionally equivalent to broadcast), may be the contents of the word itself, or may be some function of the input and the word contents. A CMOS hardware implementation of the AP design is being tested, and preliminary results indicate that some hardware revisions will be necessary. Questions of AP control have been considered, and the details will be worked out through use of a software simulation.

PROBLEM STATUS

This is an interim report on a continuing NRL Problem.

AUTHORIZATION

NRL Problem R06-41
Projects, NavAirSysCom WF08-151-702 and WF15-241-601/602

Manuscript submitted September 7, 1971.

ACRONYMS

AADC	Advanced Avionics Digital Computer
AMAR	Associative memory address register
AMOB	Associative memory output buffer
AP	Associative processor
APC	Associative processor controller
APIR	AP instruction register
APM	Associative processor memory
CB	Clock bit
CM	Clock mask
CMOS	Complementary MOS
COMP	Comparand
MOS	Metal oxide semiconductor
NRSP	Number of ROB's set previously register
RB	Response bit
RCB	Reset clock bit
ROB	Response output bit
RRB	Reset response bit
RRN	Response resolution network
RS	Response store
SC1	Search control one
SC2	Search control two
SDR	Syllable definition register

UNCLASSIFIED

ANOTHER ASSOCIATIVE PROCESSOR

1. Introduction and Summary

The design of a flexible, highly parallel associative processor has been described in a previous report (1). During the first phase of the subsequent hardware development of the associative processor (AP), the design was changed considerably. The additional flexibility and capability of the new design are the result of orienting the AP toward the requirements of the Advanced Avionics Digital Computer (AADC) under development by the Naval Air Systems Command. This report will describe the new design and will give preliminary results of the fabrication of a four-bit integrated circuit chip.

The original design is for a logic-in-memory array which may be briefly described as a bit-parallel associative processor with multifield search, arithmetic, and logical capability. Field definitions are bit variable and under software control. The AP is of the broadcast type, in that all data and controls are broadcast simultaneously to all words. This results in fast execution speed, but the lack of internal or local communication leads to application limitations. A compromise in which both broadcast and local modes were included was considered; however, it was rejected due to the additional expense associated with the required gates and control lines.

As the original design evolved, the speed associated with broadcast operation was traded for the increased capability provided by interword communications. Data flow within the AP is now a vertical trickle. Data for each word is fed in from the word above and fed out to the word below. The output to the word below may be the same as the input (functionally equivalent to broadcast), may be the contents of the word itself, or may be some function of the input and the word contents. The response store, which in the original design recorded the results of searches and provided the horizontal control inputs to each word, has been modified to take advantage of these new features. As a compromise between AP flexibility and the technological constraints of gate and pin counts the bit-variable field definition capability has been replaced with a byte-variable capability, each byte containing four bits. The evolution of this design from the original design discussed in Ref. 1 in large part reflects the work of Kautz (2,3).

The capabilities of a single AP word will be described in Section 2. The hardware implementation of the single-byte integrated-circuit chip will be described in Section 3. A description of the response store (RS) will be given in Section 4. Several simple examples of AP operations will be given in Section 5. AP control will be considered in Section 6. Future work will be described in Section 7.

Note: The work of this report was performed when both authors were with the Intercept and Signal Processing Branch, Electronic Warfare Division.

2. Word Description

The basic inputs and outputs of a single AP word are shown in Fig. 1. Each word consists of an integral number of four-bit bytes and is organized into a set of syllables or fields by the syllable definition register (SDR) inputs. The definition of syllable limits is required for comparisons, arithmetic operations, and shifting. An SDR input of "1" to a particular byte indicates that the byte is the least significant byte of a syllable. Thus, for example, carries propagating from the right (direction of lower significance) are inhibited at the beginning of this byte. A word consisting of M bytes may be divided into as many as M four-bit syllables or as few as one 4M-bit syllable.

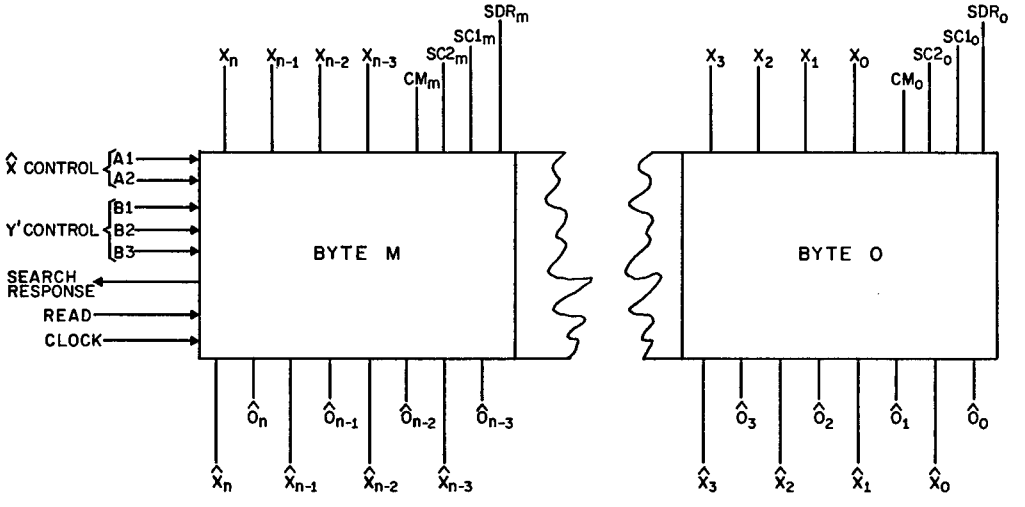

Fig. 1—A single AP word with its basic inputs and outputs

2.1 Multifield Search

The contents of a word are compared with the word inputs coming in on the X_i lines, and the result of the comparison appears on the search response line. The search control one (SC1) and search control two (SC2) inputs are used to specify search criteria on a syllable-by-syllable basis. The two lines permit the specifications of four search types: greater than or equal to, less than or equal to, exact match, and don't care.

The code for these searches is given in Table 1. The SC1 and SC2 inputs must be the same for all bytes in a syllable. If the contents of every syllable satisfy the corresponding search criteria, then the search response line indicates success. The syllable searches are logically ANDed, so that if at least one syllable search is not successful, then the search response line indicates failure.

UNCLASSIFIED

Table 1
Code for Specifying the Type of Search

Input to All Bytes		Type of Search
SC1	SC2	
0	1	Greater than or equal to
1	0	Less than or equal to
1	1	Exact match
0	0	Don't care

2.2 Control of Word Contents

If the content of a particular bit is designated by Y_i, then the state to which this will change if a clock is applied will be designated by Y'_i. There are eight possible choices for Y'_i, controlled for the entire word by the three inputs B1, B2, and B3. The eight choices and the control code for Y' selection are shown in Table 2. The contents of any byte may be preserved through use of the clock mask (CM) inputs. If a CM input is "1", the clock is not applied to the four bits in the corresponding byte and Y_i therefore does not change to Y'_i. The clock pulse, however, continues to propagate through the entire word. When either left or right shifts are clocked, information does not cross syllable lines defined by the SDR inputs.

Table 2
Code for Selecting Y', the State of a Word After a Clock is Applied

Input			Description
B_1	B_2	B_3	
0	0	0	$Y'_i = X_i$ (input data bit)
0	0	1	$Y'_i = \overline{Y}_i$ (complement)
0	1	0	$Y'_i = Y_{i-1}$ (left shift)
0	1	1	$Y'_i = Y_{i+1}$ (right shift)
1	0	0	$Y'_i = X_i \cdot Y_i$ (AND)
1	0	1	$Y'_i = X_i + Y_i$ (OR)
1	1	0	$Y'_i = X_i \oplus Y_i \oplus$ carry (sum)
1	1	1	$Y'_i = X_i \oplus Y_i$ (EXCLUSIVE OR)

2.3 Control of Word Output

The $\hat{X}_i$ are the data output lines of the word. These outputs are controlled for the entire word by the two inputs A1 and A2. There are three choices (Table 3):

1. $\hat{X}_i = X_i$ (pass),
2. $\hat{X}_i = Y_i$ (output contents),
3. $\hat{X}_i = Y_i'$. (output function).

Table 3
Code for Controlling $\hat{X}$, the Word Output

Input		Description
A1	A2	
0	0	$\hat{X}_i = X_i$ (pass; bit-slice search)
0	1	$\hat{X}_i = X_i$ (pass)
1	0	$\hat{X}_i = Y_i'$ (output function)
1	1	$\hat{X}_i = Y_i$ (output contents)

The function-output choice gives the capability to output the result of all eight logical and arithmetic operations provided by the Y' control, whether or not the word is clocked. The fourth possible input for $\hat{X}$ control, there being two control lines, is used to modify the search algorithm (Section 2.1) as follows: If A1 = A2 = 0, the output is set to $\hat{X}_i = X_i$ and the word enters the bit-slice search mode. Then if for every X_i input that is "1" the corresponding word content Y_i is also "1", the search response line gives indication of a successful search. The contents of all bits with X_i inputs that are "0" are ignored. In this mode it is possible to examine the status of individual bits within the word, a resolution not provided by the standard exact-match search that was discussed in Section 2.1.

The contents of the word may also be read out at any time through use of the READ line (Fig. 1). If this is set to "1", then the word contents are gated to the $\hat{O}_i$ lines. Providing this capability in addition to the $\hat{X}_i = X_i$ output mode significantly improves the efficiency of the AP, since it is possible to read information out and continue processing simultaneously.

3. Hardware Implementation

In the hardware implementation of the AP, each four-bit byte of the word shown in Fig. 1 consists of a single integrated circuit chip. The inputs and outputs of one such chip are shown in Fig. 2. Some of these are the complements of variables used in the discussion below. Lines that are in addition to those shown in Fig. 1 have been added to complete the functions that were discussed in Section 2. The implementation of these functions is best described by the Boolean equations that were the basis of the circuit design.

UNCLASSIFIED

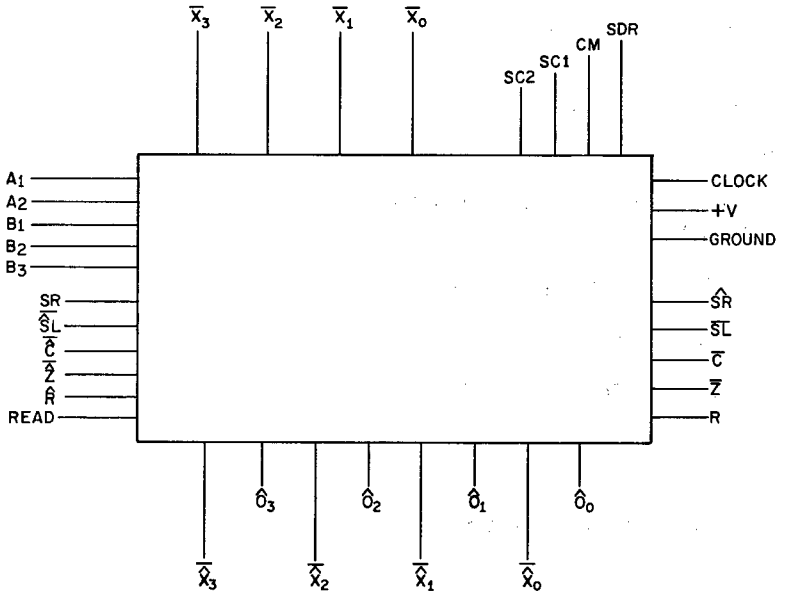

Fig. 2—A single byte, consisting of a single integrated circuit chip, with its inputs and outputs. Some of the variables are complemented.

3.1 Notation and Definitions

In the equations to follow, variables with a hat (▲) are outputs from one bit (to another) or from the chip, and variables without a hat are inputs either to a bit within the byte or to the chip as a whole. Bit variables are capital letters with subscripts, and byte variables are capital letters without subscripts. Byte variables are defined as follows:

A1, A2 = output control ($\hat{X}$),

B1, B2, B3 = function control (Y′),

C, $\hat{C}$ = carry in and out,

CLOCK = clock,

CM = clock mask ("1" inhibits CLOCK),

Z, $\hat{Z}$ = byte search response in and out ("0" denotes success),

R, $\hat{R}$ = cumulative word response in and out ("0" denotes success),

READ = readout command,

SC1, SC2 = search control lines,

S1, S2 = modified search control lines (internal variables),

SDR = syllable definition line ("1" denotes the least significant byte in a syllable),

SL, $\hat{SL}$ = left shift in and out,

SR, $\hat{SR}$ = right shift in and out.

Bit variables are defined as follows:

X_i, $\hat{X}_i$ = data input and output,

$\hat{O}_i$ = contents of bit cell storage if READ is "1",

Y_i = contents of bit cell storage (changes to Y_i' when CLOCK changes from "0" to "1" and CM is "0"),

Y_i' = one of eight functions determined by B1, B2, and B3,

C_i, $\hat{C}_i$ = carry in and out,

Z_i, $\hat{Z}_i$ = bit search response in and out.

3.2 Equations

Unless otherwise noted, the following set of equations applies to each bit in the byte (i = 0, 1, 2, 3):

$$\hat{O}_i = Y_i \cdot \mathrm{READ}, \tag{1}$$

$$\hat{X}_i = \overline{A1} \cdot X_i + A1 \cdot A2 \cdot Y_i + A1 \cdot \overline{A2} \cdot Y_i', \tag{2}$$

$$\begin{aligned} Y_i' = {} & \overline{B1} \cdot \overline{B2} \cdot \overline{B3} \cdot X_i + \overline{B1} \cdot \overline{B2} \cdot B3 \cdot \overline{Y}_i + \overline{B1} \cdot B2 \cdot \overline{B3} \cdot Y_{i-1} + \overline{B1} \cdot B2 \cdot B3 \cdot Y_{i+1} \\ & + B1 \cdot \overline{B2} \cdot \overline{B3} \cdot X_i \cdot Y_i + B1 \cdot \overline{B2} \cdot B3 \cdot (X_i + Y_i) \\ & + B1 \cdot B2 \cdot \overline{B3} \cdot (X_i \oplus Y_i \oplus C_i) + B1 \cdot B2 \cdot B3 \cdot (X_i \oplus Y_i), \end{aligned} \tag{3}$$

$$\hat{Z}_i = \overline{[\overline{S1} \cdot \overline{X}_i \cdot Y_i + \overline{S2} \cdot X_i \cdot \overline{Y}_i + \overline{Z}_i(\overline{A1} \cdot \overline{A2} \cdot \overline{X}_i + \overline{X_i \oplus Y_i})]}, \tag{4}$$

$$Z_i = \hat{Z}_{i-1} \quad (i = 1, 2, 3), \tag{5}$$

$$\hat{C}_i = X_i Y_i + C_i \cdot (X_i + Y_i), \tag{6}$$

$$C_i = \hat{C}_{i-1} \quad (i = 1, 2, 3). \tag{7}$$

The following set of equations applies to the entire byte:

$$C_0 = C \cdot \overline{\mathrm{SDR}}, \tag{8}$$

$$\hat{C} = \hat{C}_3, \tag{9}$$

$$Z_0 = Z \cdot \overline{\mathrm{SDR}}, \tag{10}$$

$$Z = \hat{Z}_3, \tag{11}$$

$$Y_4 = \mathrm{SR} \quad \text{(for use in Eq. 3)}, \tag{12}$$

UNCLASSIFIED

$$Y_{-1} = SL \cdot \overline{SDR} \quad \text{(for use in Eq. 3)}, \tag{13}$$

$$S1 = SC1 + \overline{A1} \cdot \overline{A2}, \tag{14}$$

$$S2 = SC2 + \overline{A1} \cdot \overline{A2}, \tag{15}$$

$$\widehat{SR} = Y_0 \cdot \overline{SDR}, \tag{16}$$

$$\widehat{SL} = Y_3, \tag{17}$$

$$\hat{R} = R + Z \cdot SDR. \tag{18}$$

3.3 Chip Description

The single-byte integrated circuit chip (Fig. 3) has been made by the RCA Corporation using CMOS technology. It contains 608 MOS transistors in an area of 161 by 155 mils and is mounted in a 40-pin dual-in-line package. Typical delays for the important signal paths are approximately 100 nanoseconds for $X \rightarrow \hat{X}$ and several hundred nanoseconds for $Z \rightarrow \hat{Z}$ and $C \rightarrow \hat{C}$ at 15 volts operating voltage. Production chips would be somewhat faster. A thorough engineering evaluation is currently in progress and results will be given in a separate report. Preliminary results have shown sufficient problems so that redesign of the masks will be required if production is considered. In this case the usual one-time cost of an initial custom chip may have to be paid twice.

3.4 Word Construction

AP words such as shown in Fig. 1 are constructed from single byte chips as follows: The A_i, B_i, READ, and CLOCK signals are broadcast to all chips in the word. Each $\widehat{SL}$, $\hat{C}$, $\hat{Z}$, and $\hat{R}$ output is connected directly to the corresponding SL, C, Z, and R inputs of the byte immediately to the left (more significant byte). Each $\widehat{SR}$ output is connected directly to the SR input of the byte immediately to the right (less significant byte). The C, Z, and R inputs to the first (least significant) byte are set to "0". The extreme right and left SL and SR inputs may also be set to "0" or may be tied to input devices.

3.5 Multiple Use of Chips

One motivation for designing the AP in terms of this multifunction chip was the possibility of using the chip as an off-the-shelf component in a variety of digital systems. It can be used to implement one or more comparison, correlation, arithmetic, logical, and shifting functions. For example, chips connected as a single AP word (Fig. 1) could form the heart of the arithmetic and logic section of a conventional computer.

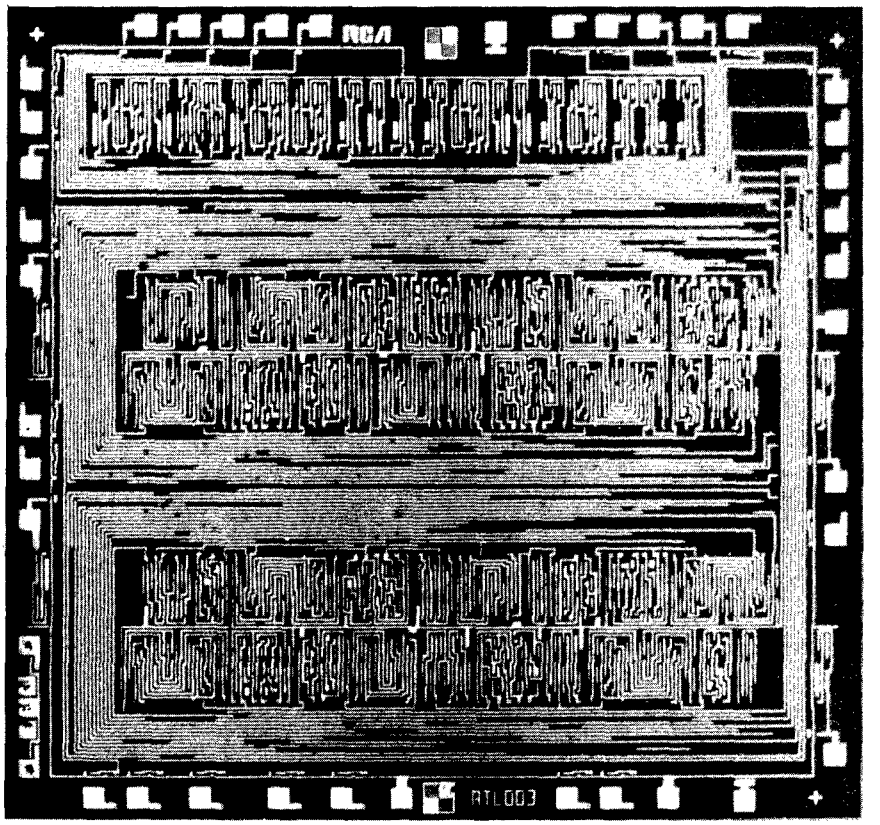

Fig. 3—Photomicrograph of a single-byte integrated circuit chip

4. System Description

4.1 Vertical Control

Basic elements of the AP are shown in Fig. 4. The associative processor memory (APM) section consists of a set of previously described AP words (Fig. 1). Data input to the APM is through the comparand register (COMP), which is connected to the X_i inputs of the uppermost word in the APM. The X_i inputs of every other word are connected to the $\hat{X}_i$ outputs of the word directly above. Four control registers provide the other vertical inputs: the syllable definition register (SDR), search control one (SC1), search control two (SC2), and clock mask (CM). Each of these registers contains one bit for each byte slice of the APM. (A byte slice consists of the same byte of every APM word.) Each bit in these control registers is broadcast directly to the appropriate input of the corresponding byte of each APM word.

The associative memory output buffer (AMOB) is a register used to output full APM words. Input to the AMOB may come from the $\hat{X}_i$ outputs of the bottom APM word or directly from any other word through use of the READ command.

UNCLASSIFIED

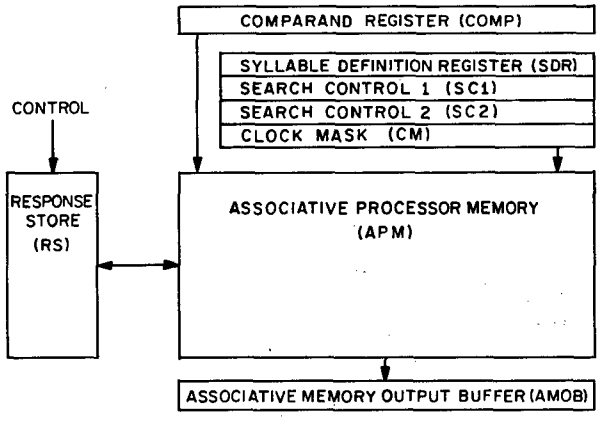

Fig. 4—Basic elements of the associative processor

4.2 Horizontal Control

Horizontal inputs to the APM ($\hat{X}$ control, Y′ control, READ, and CLOCK) are provided by a hardware section of the AP called the response store (RS). The RS also records the results of AP searches. The biggest problem in choosing an RS structure is deciding on the algorithm for setting the $\hat{X}$ and Y′ control lines for each word. If every AP instruction could set these lines independently for each word, an impractical amount of program storage would be required. In addition, although extremely complex logical functions could be performed in one trickle through the APM, the full capability would seldom be used. On the other hand, if the $\hat{X}$ and Y′ control lines were constrained to be identical for all APM words, then much of the complexity of the AP chip would be wasted. The tradeoff between these two extremes was made by considering various desirable AP operations.

A great deal of flexibility and processing power is obtained if for each instruction the $\hat{X}$ and Y′ control lines are selected from two sets. One of these, called the standard set, provides the five $\hat{X}$ and Y′ control inputs for each APM word that does not satisfy certain conditions. The other set, called the input set, provides the control inputs for those APM words that do satisfy these conditions. In the implementation of this approach the input set and the conditions for its use are provided by the AP instruction. For each instruction, one of two predefined standard sets may be specified as the standard set for this instruction through use of a single RS control line (MODE). An alternative implementation might provide both the input set and the standard set with the AP instruction. This would permit a full range of standard sets instead of the two that we have chosen. Although this would be a more powerful approach, it requires four more bits per AP instruction as well as additional RS hardware, both of which cannot yet be justified by the algorithm development work that we are engaged in.

4.3 Response Store Structure

The response store is best described in terms of the more detailed AP diagram shown in Fig. 5. Each APM word has a corresponding portion of the RS associated with it consisting of a control section and a memory section (Fig. 6). The memory section may be thought of as a continuation of the APM word. The main purpose of the memory section is to record

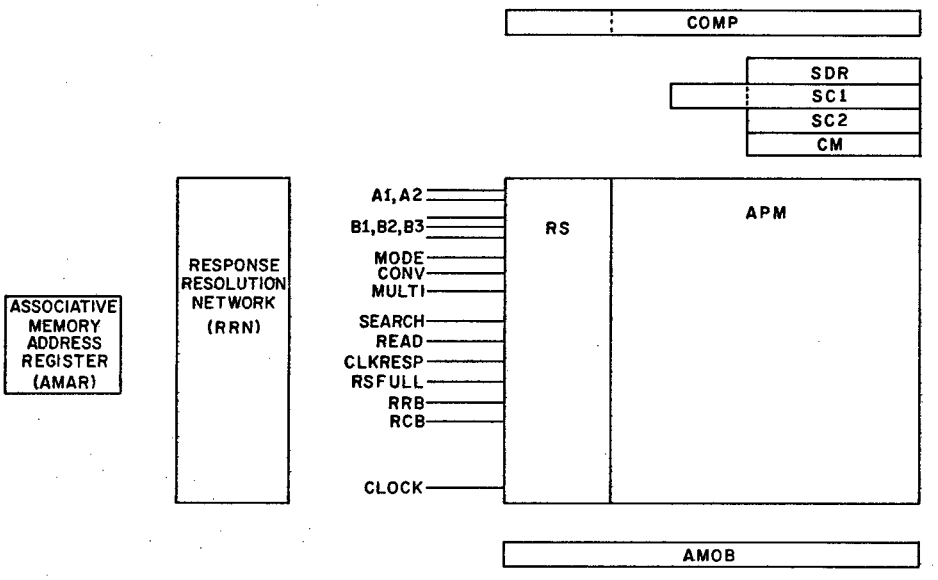

Fig. 5—Associative processor

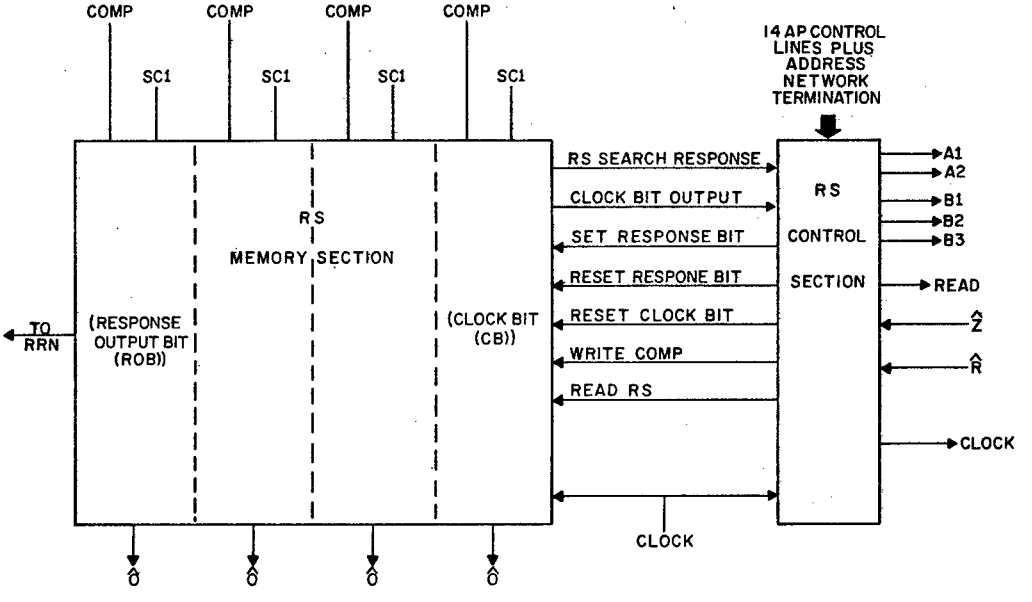

Fig. 6—Response store part of one AP word. In actual hardware implementation the control circuitry may be distributed in the memory section.

the results of different searches. Its size depends on the number of searches for which an independent record is desired. An RS memory section of four bits will be assumed in the following discussion.

It is advantageous to extend the COMP so that the RS memory section can be included in the AP search. By requiring as part of the search specification that a particular RS bit be

UNCLASSIFIED

set (or not set), the response of each word to a previous search can be made a condition of the present one. To provide for this, each bit in the extended portion of the COMP is connected directly (broadcast) to every corresponding RS bit (Fig. 6). In addition, one of the control registers, the SC1, is extended by the number of bits in RS to provide the required search control specifications. These SC1 bits are also broadcast to the corresponding RS bits. An SC1 input of "1" specifies that the corresponding COMP and RS bits must match. A SC1 input of "0" specifies that the corresponding RS bit be ignored by the RS search. The RS-search-response signal which is passed to the RS control section (Fig. 6) will indicate success if the COMP exactly matches those RS bits with SC1 inputs of "1". Bit slices that are ignored in this RS search may be used to record the result of the full word search (the combined search of the APM word and the RS memory section). An RS memory bit is specified as a response bit (RB) by "1" and "0" inputs from the COMP and SC1 respectively. This combination would not otherwise be useful. The response store will set each RB to "1" if the full word search is successful and if the SEARCH control line (Fig. 5) has been set to "1". Thus, if a search is being performed, bits in the response bit-slice (specified by the "1"/"0" COMP/SC1 pattern) will be set for those words that satisfy the full RS and APM search criteria.

When it is desired to retrieve the results of a particular search, the extreme-left-hand RS bit slice is designated as the RB. The output of this response output bit (ROB) from each word is connected to the response resolution network (RRN). The output of the RRN is the largest address of any word with a ROB output of "1". This address can be gated to the associative memory address register (AMR) upon command of the control processor. This address may then be stored elsewhere, used as a pointer to auxiliary files, or used to directly read out the AP word it references. The theory and design of the RRN, as well as a justification for this approach, has been given in Ref. 1.

While the address of a responding AP word is in the AMAR, it can be used to reset the ROB of the word it references, a necessary capability if the RRN is to be used to smoothly retrieve a list of responding addresses. The RS control section of every word has an input from the address decoding network. If the reset response bit (RRB) control line (Figs. 5 and 6) is set to "1", any RB ("1"/"0" COMP/SC1 pattern) will be set to "0" in the word whose address is currently in the AMAR.

One other RS bit, conceptually the extreme-right-hand bit in Fig. 6, has a special purpose. This bit is called the clock bit (CB) and is used by each RS control section in determining whether a clock will be applied to the corresponding word. The CB of all words will be reset to "0" on the next clock if the reset clock bit (RCB) line is set to "1".

With this background the complete RS control structure can be defined by the following outline.

I. Designation of the Standard Set of $\hat{X}$ and Y′ Controls:
   A. MODE = "0"; standard set is $\hat{X} = X$, $Y' = X$.
   B. MODE = "1"; standard set is $\hat{X} = X$, $Y' = \text{sum}$.

II. Word Selection Options (for selecting the words for a change of the standard set and for clocking):
   A. CONV = "1" and the address of the word is in AMAR.
   B. MULTI = "1", RSFULL = "0", and the RS search is successful.
   C. MULTI = "1", RSFULL = "1", and both the RS and the APM searches are successful (the full word search is successful).

III. Change of the Standard Set of $\hat{X}$ and Y′ Controls to the Input Set:
   A. If A1 ≠ "0" or A2 ≠ "0" (input set), then the standard set is changed to the input set (A1 A2 B1 B2 B3) if IIA, IIB, or IIC is satisfied.
   B. If A1 = A2 = "0" (the input set specifies the bit-slice search mode), $\hat{X}$ controls for all words are changed to the bit-slice search mode. The Y′ controls are changed from the standard set to the input set (B1B2B3) if IIA, IIB, or IIC is satisfied.

IV. Application of the clock to an APM word:
   A. A clock is applied if the clock bit (CB) is set.
   B. A clock bit is applied if IIA, IIB, or IIC is satisfied and CLKRESP = 1 (clock-on response).

V. Special Functions:
   A. The RB of an addressed word is set to "0" if the RRB line is "1".
   B. The CB of all words is set to "0" if the RCB line is "1".
   C. The RB is set to "1" if the SEARCH line is "1" and the full word search is successful.
   D. The full contents of the addressed word are read out (gated to AMOB) if the READ line to AP is "1".
   E. The RS section of COMP is written into the RS if IIA is satisfied and if the input set specified B1 = B2 = B3 = 0 (Y′ = X). This includes the RS in the conventional write operation.

The AP clock is always distributed to the RS memory sections. The RS contents will change on a clock pulse if any of the set response bit, reset response bit, reset clock bit, or write COMP lines are raised by the corresponding RS control section. The distribution of the clock to the APM word itself, however, depends on the control structure defined above.

5. Examples of AP Operations

In this section, we give three examples of AP operations that can be performed with the control structure defined in Section 4.

5.1 Search for the Word With the Greatest Contents

For convenience we shall describe the first example in terms of a single-syllable APM. The search, however, can be made on any syllable of a multisyllable APM, as was discussed in Section 2.1

Suppose the COMP contents are set to zero (except for the RS associated but used to specify the RB) and the SC1 and SC2 registers specify a greater than or equal to search. Also assume that the CB slice is zero. Let the control inputs to the AP be as follows (constituting one AP instruction):

UNCLASSIFIED

Input set: $\hat{X} = Y$,

$Y' =$ anything (since no word will be clocked);

MODE = 0 (standard set: $\hat{X} = X$, $Y' = X$);

MULTI = RSFULL = SEARCH = "1" (Change from the standard set to the input set and set RB if the full word search is successful);

CONV = READ = CLKRESP = RCB = "0".

As a result of this specification each word whose contents are greater than its X inputs will gate its own contents to the word below. Once the logic settles down, the lowest word that does so contains the largest contents of all APM words. These contents will then be presented at the $\hat{X}$ outputs of the bottom AP word, since all words below the greatest one maintain the standard set with $\hat{X} = X$. These $\hat{X}$ outputs may then be gated onto the AMOB for use as the result of the search. Alternatively, if the ROB was specified as an RB for this search, the RRN will subsequently produce the address of the AP word with greatest contents (Section 4.3).

5.2 Multiplication of Fields in Adjacent Words

For the second example we consider the two adjacent AP words in Fig. 7. The APM has been divided into three syllables. We wish to multiply the syllable in position A of every word whose RS contents are X11X (X indicates don't care) by those in positions D_1 and D_2 of the word above it (which has RS contents X10X). The results are to be stored as shown. This procedure is the basis for matrix multiplication within the AP and may be accomplished simultaneously for all such pairs of words by repeating two instructions. To begin with, a bit-slice search is performed on the first bit slice of syllable A while the RS contents are required to be X11X. The CB slice is designated as the RB. This instruction has the effect of copying the first bit of A into the clock bit of all words whose response store contents are X11X. The second instruction has the RS search criteria X10X and control lines set as follows:

Input set: $\hat{X} = Y$,

$Y' =$ left shift;

MODE = 1 (standard set: $\hat{X} = X$, $Y' =$ sum);

MULTI = CLKRESP = RCB = "1";

RSFULL = READ = CONV = RRB = SEARCH = "0".

As a result, all words with an RS pattern X10X will have their contents passed (via $\hat{X}$) to the word below and will be left shifted by one when the clock is applied. If the CB is set in the word below (the first bit in A is a "1"), then D_1 and D_2 will be added to the corresponding syllables in this word (the standard set specifies $Y' =$ sum), which are assumed to be zero initially. This instruction also resets all clock bits to zero.

The two instructions are now repeated, only this time the first instruction copies the second bit slice of syllable A into the CB slice. The second instruction will add (where appropriate) D_1 and D_2 shifted to the left by one. Repeating this sequence of two instructions for each bit in A will result in the products as shown in Fig. 7.

5.3 Column Addition

As the third example, the corresponding syllables of any set of AP words that can be identified by RS or full word search criteria may be added in a single instruction, the result appearing at the $\hat{X}$ output of the bottom AP word. The COMP, SC1, and SC2 are loaded according to the search cirteria, and the control lines are set as follows:

Input set: $\hat{X} = Y'$

$Y' = \text{sum};$

MODE = "0" (standard set: $\hat{X} = X$, $Y' = X$);

MULTI = "1";

READ = "0".

RSFULL is set according to the type of criterion that is used to identify the words being added. If CLKRESP = "0", the AP contents will remain unchanged (assuming the CB slice is zero). If CLKRESP = "1", the words contributing to the column add will be clocked. The result will be a running sum, the sum of each word and all contributing words above it. The other control lines may be set according to the functions desired. For example, one other word could be included in the addition regardless of the search criteria by placing its address in the associative memory address register (AMAR) and setting CONV = "1".

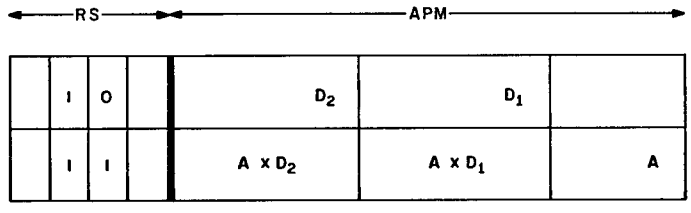

Fig. 7—Multiplication of fields in adjacent words

6. The Associative Processor Controller

The AP will be under the control of a sequential stored-program computer called the associative processor controller (APC). Figure 8 is a block diagram showing the relationship between an idealized APC and the AP. The various registers required to control the AP are special-category registers within the APC. Two other special registers have been added. The NRSP [illegible] by the RRN with the approximate number of ROB's [illegible] previous to the

UNCLASSIFIED

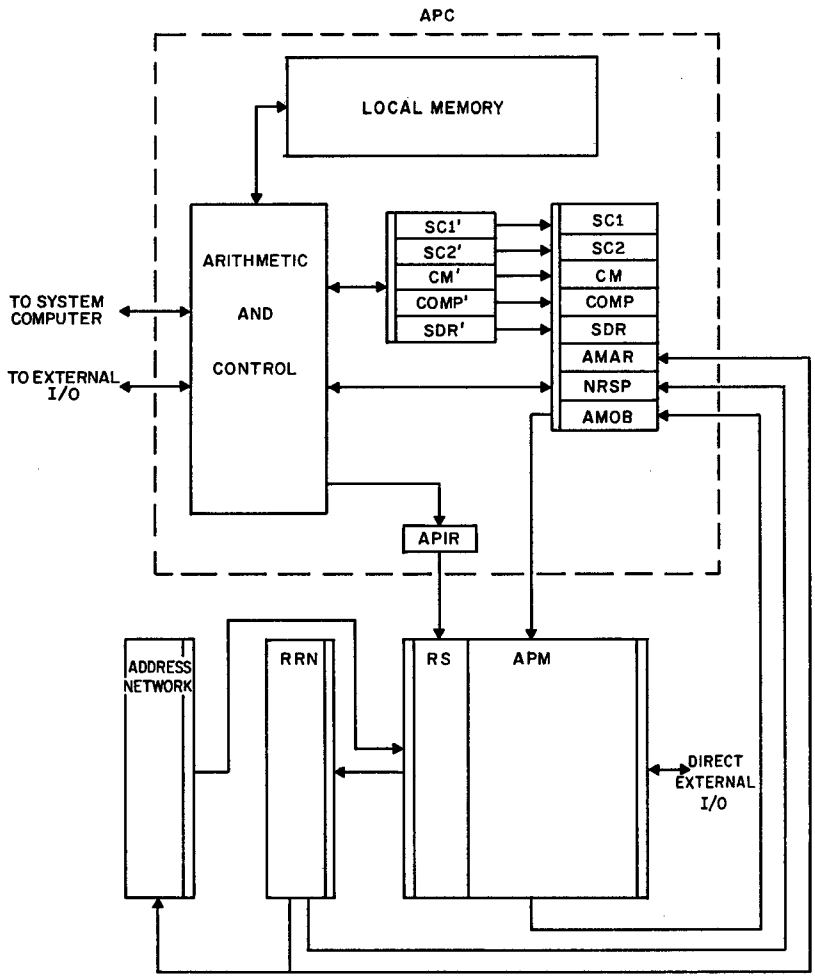

Fig. 8—Idealized combination of the AP and an associative processor controller (APC).

clock. This might be obtained by passing all of the ROB outputs through a summation amplifier and performing an analog-to-digital conversion on the result. The NRSP enables APC program branches that depend on the number of responses to an AP search. The AP instruction register (APIR) drives the 14 RS control lines. The possibility of a direct AP I/O through the SL and SR lines of the first bit slice is also shown.

The nine AP control registers (SC1, SC2, CM, COMP, SDR, AMAR, NRSP, AMOB, and APIR) remain static while the APM, RS, and RRN logic settles. In the meantime the APC will remain busy performing bookkeeping instructions, systems communications, and, most important, assembling the next set of AP control registers. For this purpose five additional registers are provided (SC1′, SC2′, CM′, COMP′, and SDR′), each corresponding to one AP control register. Since these registers, especially the COMP, are likely to be considerably wider than either the APC memory or system memory, many APC cycles will be required to fill them. The primed registers can be filled while the AP circuitry is settling down under the current set of inputs. When the APC is ready, the AP is clocked and the primed registers

may be simultaneously gated to their unprimed counterparts to provide the next set of AP inputs. The APIR is also reloaded at this time.

The details of the APC design will be primarily influenced by the size of the AP, I/O requirements, the percentage and type of bookkeeping instructions necessary in AP algorithms, and the amount and level of communication between the APC and the system computer. These questions will be investigated through use of a software simulation of the AP that has been developed for this purpose (4).

7. Future Work

If the results of exhaustive chip testing are satisfactory, a small AP (between 1000 and 8000 bits) may be constructed. It would be interfaced to a mobile, medium-scale computer which would be used both to drive the AP directly in real time processing applications and to simulate various APC designs.

8. Acknowledgment

Many of the differences between the present design and the original (1) are due to the comments and suggestions of William Kautz of the Stanford Research Institute. The authors acknowledge his help with appreciation.

REFERENCES

1. Shore, J.E. and Polkinghorn, F.A., Jr., "A Fast, Flexible, Highly Parallel Associative Processor," NRL Report 6961, Nov. 28, 1969.

2. Kautz, W.H., Pease, M.C., and Green, M.W., "Cellular Logic-in-Memory Arrays," Final Report, Contract Nonr-4833 (00), Stanford Research Institute, Menlo Park, Calif. 1970.

3. Kautz, W.H., "An Augmented Content-Addressed Memory Array For Implementation With Large-Scale Integration," J. ACM 18, 19 (Jan. 1971).

4. Shore, J.E., "Software Simulation of an Associative Processor," NRL Report 7351.

Security Classification

UNCLASSIFIED

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Naval Research Laboratory Washington, D.C. 20390	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Another Associative Processor

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

An interim report on a continuing NRL Problem

5. AUTHOR(S) *(First name, middle initial, last name)*

John E. Shore and Terry L. Collins

6. REPORT DATE December 30, 1971	7a. TOTAL NO. OF PAGES 21	7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. NRL Problem R06-41 b. PROJECT NO. WF08-151-702 c. WF15-241-601/602 d.	9a. ORIGINATOR'S REPORT NUMBER(S) NRL Report 7348 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Department of the Navy Naval Air Systems Command Washington, D.C. 20360

13. ABSTRACT

The logical design of an associative processor (AP), which has evolved from the design previously reported in NRL Report 6961, has now been oriented toward the requirements of the Advanced Avionic Digital Computer (AADC) under development by the Naval Air Systems Command. The AP is a byte-variable processor with multifield search, arithmetic, and logical capability. Data flow within the AP is a vertical trickle. Data for each word is fed in from the word above and fed out to the word below. The output may be the same as the input (functionally equivalent to broadcast), may be the contents of the word itself, or may be some function of the input and the word contents. A CMOS hardware implementation of the AP design is being tested, and preliminary results indicate that some hardware revisions will be necessary. Questions of AP control have been considered, and the details will be worked out through use of a software simulation.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Computers						
Design						
Associative storage						
Associative processors						
Parallel processors						
Integrated circuits						
Semiconductor computer storage						
Array processors						